Необхідність збільшення швидкодії і продуктивності криптографічних алгоритмів вимагає вживання нових методів. Для вирішення даної проблеми досліджується можливість вживання паралельних технологій

Ключові слова: Rijndael, розпаралелювання

Необходимость увеличения быстродействия и производительности криптографических алгоритмов требует применения новых методов. Для решения данной проблемы исследуется возможность применения параллельных технологий

Ключевые слова: Rijndael, распараллеливание

Necessity of increase in speed and productivity of cryptography algorithms demands application of new methods. For solution of the given task possibility application of parallel technologies is researched

Key words: Rijndael, parallelization

УДК 004.272.2:004.421.6.032.26:003.26

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ МИНИМИЗАЦИИ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ ПУТЕМ ИХ РАСПАРАЛЛЕЛИВАНИЯ

Г.М. Кривошеева

Аспирант
Кафедра ПО ЭВМ
Харьковский национальный университет радиоэлектроники
пр. Ленина, 14, г. Харьков, Украина, 61166
Контактный тел.: 8-099-683-49-11
E-mail: edelweyz@gmail.com

1. Введение

Современный уровень развития информационных технологий выдвигает на передний план новые требования к построению систем защиты информации и обеспечению информационной безопасности по открытым каналам связи. Прогресс вычислительных систем и растущие требования пользователей к обеспечению требуемой степени защиты информации стимулируют к разработке новых криптографических алгоритмов, которые бы позволяли эффективно провостоять атакам злоумышленников, вплоть до того момента, чтоб им проще было добыть желаемую информацию другим путем. Для обеспечения достаточной криптографической стойкости в современных условиях необходимо использовать системы с большой длиной ключа, причем длина ключа все время растет, а значит, растет вычислительная сложность криптографических операций. Время выполнения криптографических алгоритмов также очень важно, так как определяет возможность использования того или иного алгоритма в системах защиты информации. Последовательные алгоритмы, которые используются, не дают возможности для увеличения скоростей. Одним из направлений уменьшения вычислительной сложности является распараллеливание алгоритмов в системах с общей и распределенной памятью. Развитие персональных ЭВМ с многопроцессорными и многоядерными архитектурами делает возможным применение параллельно выполняемых криптографических программных средств. Эти же направления можно использовать при разработке аппаратных криптографических систем.

2. Постановка задачи

В данной работе предполагается исследовать возможность уменьшения вычислительной сложности криптографических алгоритмов путем их распараллеливания.

Существует ряд криптографических алгоритмов проверенных временем, к одному из таких алгоритмов относится Rijndael, принятый как стандарт AES.

Rijndael имеет наилучшее сочетание стойкости, производительности, эффективности реализации и гибкости. Его низкие требования к объему памяти делают его идеально подходящим для встроенных систем.

Данный алгоритм, по данным таблицы 1, имеет наибольшую способность к эффективному применению параллельных методов [3].

Таблица 1

Параллелизм и теоретическая производительность

Криптоалгоритм	Число циклов на критическом пути по данным двух источников		Максимальное число процессоров, эффективно работающих параллельно
MARS	214	258	2
RC6	181	185	2
RIJNDAEL	71	86	7
SERPENT	526	556	2
TWOFISH	162	166	3

Rijndael используется для шифрования данных в системах Rar, GnuPG, Bestrypt.

3. Описание криптоалгоритма Rijndael

Материалы данного раздела в значительной степени основываются на авторском описании алгоритма Rijndael [1] и документе FIPS -197 [2].

Rijndael – это итерационный блочный шифр, имеющий архитектуру "Квадрат". Шифр имеет переменную длину блоков и различные длины ключей. Длина ключа и длина блока могут быть равны независимо друг от друга 128, 192 или 256 битам. В стандарте AES определена длина блока данных, равная 128 битам.

Алгоритм оперирует байтами, которые рассматриваются как элементы конечного поля GF(2^8).

Промежуточные результаты преобразований, выполняемых в рамках криптоалгоритма, называются состояниями (State).

Рассмотрим подробнее операции, используемые в Rijndael.

1. Замена байтов (SubBytes). Преобразование SubBytes () представляют собой нелинейную замену байтов, выполняемую независимо с каждым байтом состояния. Таблицы замены S-блока являются инвертируемыми и построены из композиции следующих двух преобразований входного байта:

1)получение обратного элемента относительно умножения в полосе GF(28), нулевой элемент {00} переходит сам в себя;

2)применение преобразования над GF(2), определенного таким образом:

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{bmatrix} + \begin{matrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{matrix}$$

2. Преобразование сдвига строк (ShiftRows). Последние 3 строки состояния циклически сдвигаются влево на различное число байтов. Строка 1 сдвигается на C_1 байт, строка 2 – на C_2 байт, и строка 3 – на C_3 байт. Значения сдвигов C_1, C_2 и C_3 в Rijndael зависят от длины блока N_b. Инверсией для ShiftRows является циклический сдвиг трех нижних строк соответственно на $N_b - C_1$, $N_b - C_2$ и $N_b - C_3$ байт, чтобы байт в позиции j в строке i перемещался в позицию $(j + N_b - C_i) \bmod N_b$.

3. Преобразование перемешивания столбцов (MixColumns). В этом преобразовании столбцы состояния рассматриваются как многочлены над GF(28) и умножаются по модулю $x^4 + 1$ на многочлен g(x), выглядящий следующим образом:

$$g(x) = \{03\}x^3 + \{01\}x^2 + \{02\}.$$

Данный полином является взаимнопростым с $x^4 + 1$ и, следовательно, инвертируем, поэтому операция MixColumns обратима.

4. Добавление раундового ключа (AddRoundKey). В данной операции раундовый ключ добавляется к состоянию посредством простого поразрядного XOR. Раундовый ключ вырабатывается из ключа шифрования посредством алгоритма выработки ключей (key schedule). Длина раундового ключа (в 32-разрядных словах) равна длине блока N_b.

Алгоритм выработки ключей (Key Schedule)

Раундовые ключи получаются из ключа шифрования посредством алгоритма выработки ключей. Он содержит два компонента: расширение ключа (Key Expansion) и выбор раундового ключа (Round Key Selection). Основополагающие принципы алгоритма выглядят следующим образом:

- общее число битов раундовых ключей равно длине блока, умноженной на число раундов, плюс 1 (например, для длины блока 128 бит и 10 раундов требуется 1408 бит раундовых ключей);
- ключ шифрования расширяется в расширенный ключ (Expanded Key);
- раундовые ключи берутся из расширенного ключа следующим образом: первый раундовый ключ содержит первые N_b слов, второй – следующие N_b слов и т. д.

Расширенный ключ в Rijndael является линейным массивом четырехбайтных слов и обозначается как W [N_b (N_r +1)]. Первые N_k слов состоят из ключа шифрования. Остальные слова определяются рекурсивно. Функция расширения ключа N_k : существует версия функции для Nk , равным или меньшим 6, и версия для N_k больше 6.

4. Возможность распараллеливания алгоритма Rijndael

Наиболее распространенными подходами к распараллеливанию вычислений и обработки данных являются подходы, основанные на моделях параллелизма данных и параллелизма задач. В основе каждого подхода лежит распределение вычислительной работы между доступными пользователю процессорами параллельного компьютера. При этом приходится решать разнообразные проблемы. Прежде всего, это равномерная загрузка процессоров, т. к. если основная

вычислительная работа будет ложиться только на часть из них, уменьшится и выигрыш от распараллеливания. Другая не менее важная проблема — эффективная организация обмена информацией между задачами. Если вычисления выполняются на высокопроизводительных процессорах, загрузка которых достаточно равномерна, но скорость обмена данными при этом низка, большая часть времени будет тратиться впустую на ожидание информации, необходимой для дальнейшей работы задачи. Это может существенно снизить скорость работы программы.

В алгоритме Rijndael заложена высокая степень параллелизма. Все четыре основные операции могут работать параллельно либо с несколькими байтами (функция SubBytes () и AddRoundKey()), либо со строками массива State (функция MixColumn()).

При использовании T – таблиц все четыре обращения к ним можно также осуществлять параллельно. Операция сложения по модулю два легко распараллеливается.

Алгоритм развертывания ключа имеет последовательный характер: значение w[i] напрямую зависит от w[i-1]. Однако в приложениях, где скорость является критическим параметром, развертывании ключа, как правило, делается единожды для большего числа преобразуемых данных. Там же, где ключ меняется часто (в пределе – каждый раз для каждой новой порции данных), операции развертывания ключа и преобразования данных можно производить одновременно, т.е. параллельно.

Алгоритм Rijndael является блочным шифром, следовательно, наиболее простой способ применения параллельных методов – это обработка каждого блока на отдельном процессоре.

Наиболее рациональным способом обработки будет использование подхода, основанного на моделях параллелизма данных. К плюсам данного варианта относится простота реализации и быстрая обработка, т.к. нет частого обращения к файлу данных. Недостатками данного варианта является потребность в больших ресурсах ОЗУ, а также слабая синхронизация вычислений на параллельных процессорах, т. е. выполнение команд на разных процессорах происходит, как правило, независимо и только иногда производится согласование выполнения циклов или других программных конструкций — их синхронизация. Каждый процессор выполняет один и тот же фрагмент программы, но нет гарантии, что в заданный момент времени на всех процессорах выполняется одна и та же машинная команда.

5. Выводы

Постоянный научно-технических прогресс в области информационных технологий, в частности в области, относящейся к защите информации, требует разработки новых методов для увеличения быстродействия и устойчивости к взлому криптографических алгоритмов. Наиболее эффективным решением данной проблемы является применение параллельных технологий. На данный момент развития информационных технологий слабо разработаны теоретические методы и модели распараллеливания алгоритмов вообще и криптографических алгоритмов в частности.

Для блочных шифров, таких как Rijndael, наиболее простой способ применения параллельных методов – это обработка каждого блока на отдельном процессоре.

Наиболее рациональным способом обработки будет использование подхода, основанного на моделях параллелизма данных.

Анализ и исследование эффективного использования параллельных технологий планируется провести в последующих работах.

Литература

1. Daemen J., Rijmen V. AES Proposal: Rijndael. http://csrc.nist.gov/encryption/aes/.
2. Federal Information Processing Standards Publication. Announcing the Advanced Encryption Standard (AES). http://csrc.nist.gov/encryption/aes/index.html.
3. Стандарт криптографической защиты – AES. Конечные поля / Под ред. М. А. Иванова – М.: КУДИЦ-ОБРАЗ, 2002. – 176 с.
4. Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем. СПб.: БХВ-Петербург, 2002.